Dr. Browse, A Digital Image File Format Browser

Alan H. Rowberg and Thurman Gillespy, 3rd.
Department of Radiology
University of Washington
Seattle, WA

The emerging widespread adoption of the Digital Imaging Communications in Medicine (DICOM) standard will increase the demand for radiologic image transfer between radiologic image acquisition, archive, display and printing devices. Unfortunately, there are and will continue to be many devices that do not and will not support this standard, especially older radiologic equipment and devices from non-radiologic vendors. Determining the image file format characteristics of images from such equipment is often difficult, and done on an ad hoc basis. We have developed a software tool that assists users in determining the image file format parameters of unknown radiologic images.

BACKGROUND

Discussions of data formats in both the engineering and the radiology literature have included descriptions of formats which are proposed as standards [1-4] or formats used by specific software [5-7], and reviews of commonly available image formats [8-11]. More recently, the American College of Radiology and the National Electrical Manufacturers Association (ACR-NEMA) have developed and published the DICOM standard [12-17], which is being embraced by the majority of radiology equipment manufacturers for their new equipment. Unfortunately, little has been written about how to determine if a given digital image is in one of the common formats, in a format which can be easily decoded, or in a format so cryptic that decoding it will require a great deal of effort. The typical imaging scientist is motivated by specific problems, and applies techniques to a given unknown image until either that image format is adequately understood or the problem is viewed as not having a practically reachable solution. In the past there has been little impetus for anyone to explore the general process of image format determination in a logical manner.

METHODS

To determine the file format of unknown radiologic images, the following steps are suggested:

1. Obtain any *a priori* information.

In most circumstances one can get the image matrix size (horizontal and vertical image axis dimensions) and other information, such as pixel size, image location, patient name, and other demographics from the scanner. The filename itself may provide clues about the file format. While the name may contain an identifier such as an exam number, it may also contain information about the format, such as a TIFF file which has the extension .TIF or a Targa file with .TGA as its extension.

A photograph of the image made on the scanner will usually contain much of this information, in addition to showing the nature of the image and the anatomy involved.

2. Examine file sizes of related files.

In a situation where one has a group of related image files with little or no information about the file format, it is usually best to start with a list of a group of related files and compare their sizes. Image file sets from a CT exam without data compression may begin with a file of archive or patient identifiers. Next there may be one or more files of localizer images. Then there will be a series of image files, usually of equal length and all in the same format.

Changes in file length may be due to compression, changes in image size, or to different image types, such as localizer and axial images.

3. Is the file compressed?

Images which are already compressed will show little or no additional compression when processed by any standard compression program (e.g. StuffIt, PKZIP or UNIX compress). The headers of such image files are often not compressed, so some additional compression may be achieved. Unfortunately, images which are compressed can rarely be "decoded" without very specific information about the file format and method of compression. Fortunately, many imaging modalities allow images to be stored in both compressed and non-compressed formats, so the

0195-4210/94/$5.00 © 1994 AMIA, Inc.

image might be obtained again, this time in a non-compressed format.

4. Estimate the header size.

The header size is estimated by multiplying the horizontal matrix size by the vertical matrix size and an estimate of the number of bytes per pixel. Often subtracting this number from the file size gives the exact length of the header for the image. Usually the entire header is in the file before the image and there is no trailer record after the image.

Header size = File size - (v * h * n)

where v = vertical image dimension, h = horizontal image dimension, and n = number of bytes per pixel (usually 1 or 2, but can be higher).

For many images, these steps alone will be sufficient for deriving enough information to display the image. Typically, an image display program can import and display the image after the information on matrix size, bytes per pixel, and size of header is entered into the program.

5. View the header with an ASCII editor.

Headers do not usually change in size from image to image (unless variable length fields such as comment fields are present.) The file can be examined with a hex/ASCII editor or file viewer and the location and nature of many fields may be obvious, since the contents of some fields (such as patient name) are known from the process of doing the scans and printing the images.

Often one will recognize a common image format like TIFF and others, especially if the image source is a peripheral device (not a scanner) such as a teleradiology system. TIFF files begin with either the two characters II or MM. A file which is based on the ACR-NEMA version 2 format will have the version name in ASCII in the file.

DR. BROWSE SOFTWARE TOOLS

The first tool is an ASCII browser. It will look for ASCII text in the file and display the text, in a manner simpler than any hex editor.

One mode gives a dense display of all text and another gives the address of each string, in hex and decimal.

The tool is able to drop 1 and 2 character words, which are usually extraneous characters, and words over 20 characters, which are most likely strings of a compressed image.

Header items which are ASCII are easy to locate and decode. Many are in ASCII, even for decimal/floating point values such as pixel size. Integer values are a little more difficult to find, but a data set with judicious variation of one value at a time will help. For instance, a series of slices with the slice number changing by one for each image will allow one to locate the slice number relatively easily.

Floating point values are usually conspicuous by a repetitive pattern every 4th byte. While the ieee floating point format is common, decoding other formats will require knowledge of which computer is used on the scanner.

Date formats also may follow a standard format, often 4 bytes for date/time, but may be in other formats.

The second tool is the Image Browser, which allows the user to examine any image file, regardless of the nature of the file, and get useful information, if not a viewable image.

The file is first displayed with an estimate of the image matrix size and header length based on the file size. This is chosen from a set of common matrix sizes, so as to minimize the number of characters left over, which are assigned to the header.

A default window width and window center are set, and they can be adjusted by the user. Alternatively, they can be calculated from the minimum and maximum values in the image and set so that the window will cover the range

WW = max - min + 1

WC = min + WW/2

where WW is the window width and WC is the window center.

The user can change WW and WC as desired. These may also be adjusted using alias names of Brightness and Contrast, but if the user invokes using the alias, the sign of changes in brightness is the opposite of WL. Contrast gets greater as WW gets smaller.

Then the user can change the image width and view the result. If the file is compressed, it will look very random. If it is uncompressed then the user can change the width, either by typing in a new value or using arrow keys to change it one pixel at a time. If the image is larger than the screen the user can pan within the image.

The user can then change the estimate of header size. Typically this is done first using the left arrow key to align the left edge of the image, then the UpArrow to skip one line at a time to align the top of the image. Finally, the bottom of the image can be adjusted to determine if there is a trailer or residual buffer after the image.

Thus the image area can be determined using an interaction with visual feedback, essentially a visual scroll.

The third tool calculates histograms of the image values. If the image format is not immediately apparent then one can do histograms on single bytes, pairs of bytes or triples. Peaks in these histograms will suggest first-order difference compression, or flag values which have special meaning in the format.

RESULTS

Dr. Browse was evaluated on nine unknown image sets, including the Elscint CT, GE CT Advantage, GE Windows Workstation, Hitachi CT, Picker CT, Picker MRI, Philips MRI, and two Fuji Computed Radiography formats. In all cases the alphabetic portions of the headers were located immediately, but it took some effort to conclusively identify some of the numeric items. Four of the formats were decoded essentially immediately, and two were decoded after just a few minutes of exploration. The remaining three were compressed formats, and were decoded only after several hours effort, although Dr. Browse revealed that they were compressed when it first displayed the images.

DISCUSSION

While many aspects of decoding an image format are much easier using these automated techniques, the power of the technique is particularly apparent when looking at an uncompressed image format with 2-byte pixels. Typically, the values are around 1000, so one byte is approximately 4 (4 times 256) and the next byte is a nearly random number. If one simply examines these bytes with a hexadecimal editor it is difficult to determine which low-order byte corresponds with which high-order byte. If the image starts on an even byte in the file instead of an odd byte, it may mimic the difference between low-byte first vs high-byte first formats (little endian vs. big endian). While the appearances of the files seen in a hexadecimal editor screen may be similar, the differences when viewed as images are dramatic.

If one is examining the bytes in the wrong order, either because the header length is off by one byte or because the byte order is wrong, the image may appear to be very noisy, having roughly an equal number of black and white values, with very little grayscale visible, yet show some evidence of the overall structure of the image, especially at high-contrast edges. If this is corrected inappropriately, the image will become recognizable but appear very similar to a first-order difference image. The vertical edges of the organs will be surrounded by black or white arcs, because of errors in whether the high order or low order byte are processed first. The user can rapidly change the byte order or change the header length by 1 byte, which will instantly correct the errors in the image.

Some programs offer some of the features of Dr. Browse, but with somewhat less convenience. Image, a public domain program produced at NIH, offers the ability to set horizontal and vertical image size, as well as header size, but is available only for the Macintosh computer. KBVision (Amerinex Artificial Intelligence Inc, Amherst, MA) is a commercial program with similar capabilities, but runs only on UNIX systems, such as Sun or Silicon Graphics.

CONCLUSIONS

While one can decode most image formats using a standard disk file utility program, the use of a specialized program, like Dr. Browse, makes the process much faster, simpler, and more reliable. Information on program availability is available from the AMIA forum on CompuServe (GO MEDSIG) or using Mosaic to read www.rad.washington.edu or by e-mail to rowberg@u.washington.edu.

References

1. Todd-Pokropek A, Cradduck TD, Deconinck F. A file format for the exchange of nuclear medicine image data: a specification of Interfile version 3.3. Nucl Med Commun 1992 Sep. 13(9). P 673-99.

2. Ando Y, Hashimoto S, Ohyama N, Inamura K. Current status of Image Save and Carry (IS&C) standardization. Comput Methods Programs Biomed 1992 May. 37(4). P 319-25.

3. Cradduck TD, Bailey DL, Hutton BF, deConninck F, Busemann-Sokole E, Bergmann H, Noelpp U. A standard protocol for the exchange of nuclear medicine image files. Nucl Med Commun 1989 Oct. 10(10). P 703-13.

4. Wicks DAG, Barker GJ, Plummer DL, Edited by: Lemke HU, Rhodes ML, Jaffe CC, Felix R. A general image file format. Computer Assisted Radiology. Proceedings of the International Symposium, CAR '91 pp. 471-6. 1991.

5. Noz ME, Maguire GQJr. QSH: a minima but highly portable image display and handling toolkit. Computer Methods and Programs in Biomedicine. vol.27, no.3. pp. 229-40. Nov.-Dec. 1988.

6. Young IR, Ling DHO, Bell DA, Edited by: Adlassnig KP, Grabner G, Bengtsson S, Hansen R. MIDAM-a generalised system for integrating medical images and patient records. Medical Informatics Europe 1991. Proceedings. pp. 174-8. 19-22 Aug. 1991.

7. Englmeier KH, Fink U, Hilbertz T. Visualization of multimodal image information in medicine. Proc Annu Symp Comput Appl Med Care. 1992. P 25-9.

8. Graef GL. Graphics formats. BYTE. vol.14, no.9. pp. 305-6, 308-10. Sept. 1989.

9. Maguire GQJr, Noz ME. Image formats: five years after the AAPM standard for digital image interchange. Med Phys. 1989 Sep-Oct. 16(5). P 818-23.

10. Wiltgen M, Gell G, Schneider GH. Some software requirements for a PACS: lessons from experiences in clinical routine. International Journal of Bio-Medical Computing. vol.28, no.1-2. pp. 61-70. May-June 1991.

11. Baumann P. Edited by: Abel D, Ooi BC. Database support for multidimensional discrete data. Advances in Spatial Databases. Third International Symposium, SSD '93 Proceedings. pp. 191-206. 1993.

12. Wang Y, Best DE, Hoffman JG, Horii SC, Lehr JL, Lodwick GS, Morse RR, Murphy LL, Nelson OL, Perry J. et al. ACR-NEMA digital imaging and communications standards: minimum requirements. Radiology. 1988 Feb. 166(2). P 529-32.

13. Seshadri SB, Khalsa S, Arenson RL, Brikman I, Davey MJ. An image archive with the ACR/NEMA message formats. Proceedings of the SPIE - Medical Imaging II. vol.914, pt.B. pp. 1409-15. 1988.

14. Horii SC, Hill DG, Blume HR, Best DE, Thompson B, Fuscoe C, Snavely D. An update on American College of Radiology-National Electrical Manufacturers Association standards activity. J Digit Imaging. 1990 Aug. 3(3). P 146-51.

15. Bidgood WDJr, Horii SC. Introduction to the ACR-NEMA DICOM standard. Radiographics. 1992 Mar. 12(2). P 345-55.

16. Best DE, Horii SC, Bennett W, Thomson B, Snavely D. Review of the American College of Radiology--National Electrical Manufacturers' Association standards activity. Comput Methods Programs Biomed. 1992 May. 37(4). P 305-9.

17. Horii SC, Bidgood WDJr. PACS mini refresher course. Network and ACR-NEMA protocols. Radiographics. 1992 May. 12(3). P 537-48.